Machine Learning Models for Jet Noise Analysis

Undergraduate Honors Thesis

Presented in Partial Fulfillment of the Requirements for Graduation with Honors Distinction in the Department of Aerospace Engineering at The Ohio State University

By

Madhav Shah

April 2019

Under the Advisement of Dr. Datta Gaitonde

Abstract

Jet noise has been an active area of research since the inception of jets and although understanding of the mechanisms behind jet noise are more understood, there is still a lack of knowledge to predict jet noise from first principles. The extraction of jet noise from flow data is a difficult process, but recently, a method has been developed that works well on computational fluid dynamics (CFD) data. This method, Doak's Decomposition, extracts the relevant pressure perturbations to get the acoustic response of the jet. The method involves solving a differential equation however, which makes it unusable on a point by point basis, which limits its application to CFD results only. The goal of this project is to use machine learning to learn the correlations between various jet flow parameters, including velocity, density and pressure perturbations, and the acoustic response of the system. Machine learning models were trained on the acoustic component, which was extracted from well validated CFD data using Doak's Decomposition, to see if it could learn to output the acoustics on a point by point basis. The machine learning was implemented using open source libraries for Python and the results of this project will further the understanding between flow parameters and the acoustics of jets.

Acknowledgements

This project would not have been completed without the help of many people throughout the process. I would first like to thank my advisor, Dr. Datta Gaitonde, for providing me with the opportunity to see how academic research is conducted and also for his constant support. Without his guidance, I would not have had this tremendous learning opportunity, and would not have been able to come as far as I have this last year. I am extremely grateful for his mentorship and support

I would also like to thank Ashwati Nair for taking the time to explain the intricacies behind machine learning and how to apply it effectively for this project. Without her help, this project would have been stuck in the planning stage on how to implement machine learning effectively for much longer. I thank her for taking the time to help me.

I owe many thanks to Surya Chakrabarti for explaining jet noise as well as showing me the ropes on using the various flow visualization tools used for this project. He also explained how to manipulate the jet data so that it would be usable for this project. My understanding of jet noise would be much less without his patience and countless explanations.

There are many people in the lab who also played a part in making sure that I was able to complete this project and I would not have been able to do it without their help. I thank everyone in the lab for their guidance on how to approach various problems.

Finally, I would like to thank my family and friends for supporting me throughout all my endeavors in life. I would not be here without them and I am thankful for their support.

Contents

List of Figures

1 Introduction

1.1 Introduction to Jet Noise

Jets have existed for over 60 years, and despite the research that has been conducted in that time, the precise prediction of jet noise from first principles is still unachieved [11]. Jet noise has been an active area of research for a variety of reasons. For military aircraft, the close proximity of crew members to these jets is hazardous to their hearing which can lead to long term health complications. Also, in stealth aircraft, jet noise remains a problem for ensuring that these aircraft remain undetectable. The problems are not limited to military aircraft. Noise pollution is a problem for communities that are located close to airports, where it disrupts the day to day activities of people in these neighborhoods, and in an effort to lower disruption, commercial flight times are restricted.

Figure 1 below, adapted from [11] is a depiction of a typical jet that exhausts to ambient conditions and its acoustic field. The flow shears against the static air which generates a shear layer that grows downstream of the jet. The potential core is a region of laminar flow while the flow in the shear layer is turbulent and has developing vortical structures within. Early in the shear layer, there are fine-scale turbulent structures. As the shear layer grows, coherent, large scale vortical structures form. The interactions within the turbulent region and these structure are the main contributors to jet noise [11].

The acoustic radiation that is generated is directive, having a higher intensity along certain shallow angles behind the jet, labeled downstream radiation in the figure above. The lower intensity acoustics generated by the jet, associated with the fine scale structures are radiated everywhere as indicated in Figure 1. Jet noise has been linked to these turbulent structures but the mechanism by which the acoustic radiation is generated is the question of interest. Acoustic waves are pressure perturbations but only components of the pressure perturbations that exist in jet flow

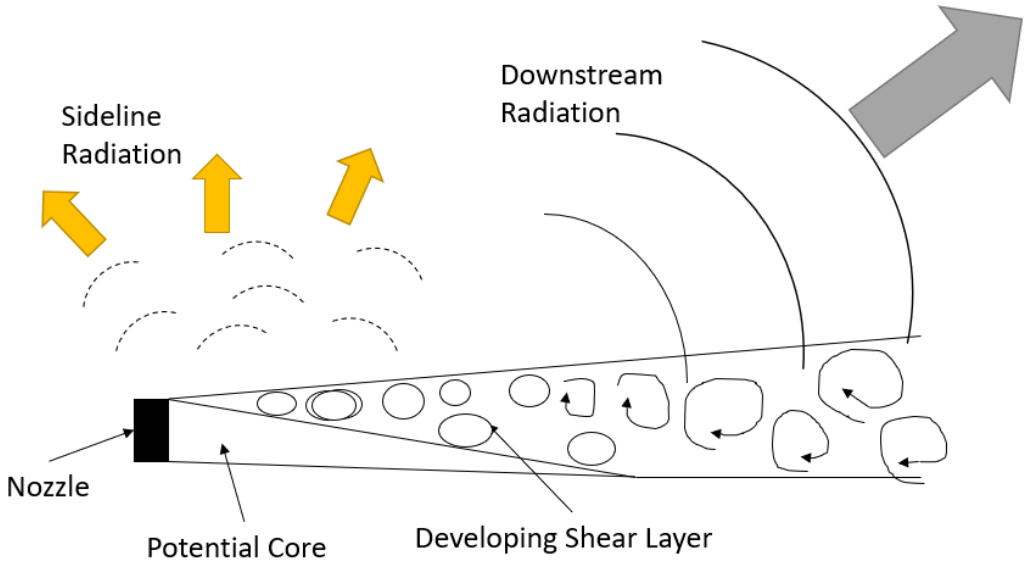

Figure 1: Jet and Acoustic Radiation

are actually acoustic waves, and though methods have recently been developed for extracting acoustics from pressure perturbations, the method requires fidelity that is present only in CFD simulations and cannot be applied to experimental data, or on a point by point basis. Developing a correlation between variables that can be measured experimentally and the acoustic response of the jet is of interest.

1.2 History of Jet Noise Analysis

Lighthill's paper in 1952 laid the groundwork for acoustic theory and present understanding follows from his work. Lighthill described the propagation of acoustics in two steps. The first is that internal stresses in a flowing fluid interact with a fluid at rest, generating noise [8]. The second step dictates that sound propagates at the speed of sound in the stationary fluid, influenced by fluctuating sources in the flowing fluid. The sources are quadruples in nature, meaning that they emit strongly in 4 directions and weekly in the others [8]. His propagation relied on building a wave propagator from the Navier-Stokes equations, with the quadruple sources,

which incorporate fluid stress and turbulence, being the strongest generator of noise [8],[11]. At the time, turbulence was thought to be purely stochastic in nature so the variables for the wave propagator were computed under this assumption. This solution was able to accurately capture the sideline acoustics but were unable to reconcile the downstream radiation, due to the previous assumption. This discrepancy pointed to coherent structures within turbulence, indicating that turbulence wasn't completely stochastic in nature. His theory has been the basis of work in acoustics that has been done since then.

Following Lighthill's work, coherence in turbulent flows was suspected. In 1971, evidence of coherent structures in the turbulent regions of jets was found, which verified the belief that turbulence is not completely stochastic in nature [4]. After these coherent structures were verified, work was done to understand the creation of these structures in a flow and how they developed. These structures propagate downstream of the jet and don't dissipate till around or beyond a jet diameter downstream [13]. As sources of jet noise, these large scale turbulent structures are non-compact sources, sustained for relatively large length scales, while fine scale turbulence is a compact source [13].

Although acoustics waves are pressure perturbations, not all pressure perturbations in a flow relate to acoustics. Perturbations, which are deviations in flow from the mean conditions, and perturbation energy can be broken into three modes: hydrodynamic, acoustic and thermal. Kovásznay found that acoustics modes are connected to density fluctuations, and the modal impacts on one another are negligible in flows with low spatial gradients [14]. However, in flows with steep gradients, the inter-modal effects become larger, and to gain a full understanding of the acoustic mode, inter-modal quadratic-coupling terms should be considered, which makes analysis more difficult. An example of these inter modal effects is when coherent vortices can create fluctuations in the thermal mode as well as scatter acoustic waves [14]. These inter modal effects can obviously make distinguishing between the modes and

their causes challenging.

The fine scale turbulence exerts a turbulence pressure on the flow, and when there are fluctuations in this turbulence pressure due to fluctuations in turbulent kinetic energy, non-directive noise is generated, which in Figure 1 is the sideline acoustic radiation [12], [13], [11]. In supersonic jets, some coherent turbulent structures convect at supersonic speeds downstream, which can be thought of as a high speed wavy wall, and generate mach wave radiation, thus giving rise to the strongly directional acoustic radiation, labeled downstream radiation in Figure 1 [13]. The process of the generation of downstream radiation in subsonic jets is slightly more complicated as the full coherent structures structures don't move at supersonic speeds. However, due to the movement and rotations of these structures, parts of them move at supersonic speeds, which is the source of the downstream radiation [13].

A method was recently developed in order to extract the acoustics from flow data. The method uses Doak's Momentum Potential Theory. The process to extract the acoustic component of the perturbation energy is done by solving a Poisson equation for density fluctuations with time [14]. This process works well for CFD simulation data since the quantities of the pressure over the whole computational domain are known, and the differential equation can be solved using the jet boundary conditions. This process can't be used for experimental data however since data over the whole experimental domain is not known.

1.3 Why Study Jet Noise?

There are many reasons why jet noise has been a large area of research for the last 60 years. Jets today, especially military aircraft, produce noise levels that hover around 150 dB (decibels). Figure 2, taken from [2], shows the noise levels of various common occurrence on the decibel scale as well as that dangers associated with different noise levels to help put this sound level into perspective. With certain military aircraft reaching 150 dB, the hearing damage to people who work on and

around the jet during operation is a real concern [3]. Even with state of the art hearing protection, the Occupational Safety and Health Administration advises that the maximum exposure time should be limited to 8.9 seconds per day to avoid damage, which can include hearing loss and tinnitus [3]. Passenger aircraft also operate around 140 db, so the problem is not limited to military aircraft. Since some airports are built near residential areas, noise pollution is of concern and flight schedules must be carefully planned to limit the affect on the residential population. A more fundamental benefit of understanding jet noise would be to increase present understanding turbulence. Turbulence, which is flow that is not as organized as laminar flows, is present in many aerospace applications including jets, helicopters, rockets, turbines etc. Despite its prevalence, turbulence is not fully understood. Understanding the mechanisms by which turbulence gives rise to acoustic waves would help further understanding in this key area.

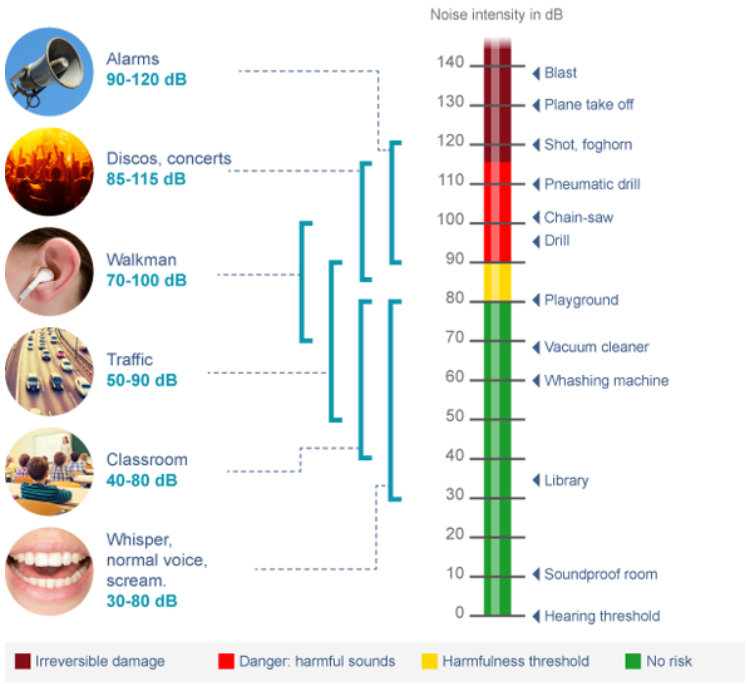

Figure 2: Decibel Scale

1.4 Significance of Current Work

The goal of this project is to analyze jet noise data on which Doak's Decomposition has been applied to see if new correlations can be developed between flow parameters and the acoustic output of Doak's. As mentioned before, Doak's Decomposition requires CFD data since the extraction of the acoustic component involves solving a differential equation, which requires a fine grid of data points. This resolution is not available in experimental setups so generating a method to generate the acoustics from point wise data is of interest. Another point of interest is lowering the computational requirements to extract the acoustic response, since numerically solving a differential equation over a large grid is computationally intensive. The data will be analyzed using machine learning techniques to learn the relationship between flow parameters and the output of Doak's Decomposition. A brief overview of machine learning and its applications to physics research is given in the next section.

1.5 Evolution of Machine Learning and its Applications

Machine learning (ML) has been growing in popularity as a tool for various tasks since its inception. Machine learning is different from typical data analysis techniques in that the algorithm isn't explicitly coded to do a specific task. The first learning algorithm was created in 1952, in which a machine learning algorithm was able to improve its performance at checkers just by playing many games, and not being explicitly given the algorithms to win [9]. This opened a door, since now, programs could be written to do tasks that they weren't explicitly programmed to do. Since then, algorithms have gotten much more complex, and are able to find correlations between various parameters by analyzing large amounts of data for many problems. This capability makes machine learning an alluring tool for physics and specifically fluids problems, since large amounts of data are generated from CFD and other computational tools.

Recently, machine learning has been effectively used to learn physical interactions from the large amount data that does exist. A team in Europe was effectively able to train a neural network in order to predict energies of molecules to a higher degree of accuracy than first order models [5]. Similar success has also been realized in the area of fluids. In the area of incompressible flow and heat conduction, an ML model was able to be trained to predict with a high degree of accuracy what the steady state solution to a boundary value problem would be after the transient phase, specifically for temperature, pressure and velocity distributions [6]. The benefit of using machine learning over the traditional solver to achieve this was that the ML algorithm was over an order of magnitude quicker in generating the solution.

In turbulent flow, a team was able to use machine learning to achieve promising results for flow control for 2D flow around a cylinder. The ML algorithm was able to suppress the strength of the Kármń vortex street and lower the drag by allowing an ML algorithm to control the jets used for flow control [10]. These results show that ML algorithms can be successful even in non-linear turbulent flows, which jet noise arises from. The specific ML algorithm of interest for this project is a multilayer perceptron (MLP), which has previously been used for fluid characteristic predictions. In 2016, an MLP network was able to predict the viscosity of nanofluids from a few relevant parameters to a high degree of accuracy [7]. This examples shows the viability for the application of MLP networks in fluids

1.6 Why Use Machine Learning?

The above example for the steady state boundary value problem shows that ML can be used for various flow problems, to achieve results that are similar to traditional solvers at a lower time and computational cost. The application of ML for flow control around a cylinder indicates that neural networks can learn the non-linearities that arise in turbulent flows, which is important since jet noise arises from the turbulence within the shear layer. Also, machine learning algorithms were able to achieve accurate results with a time advantage against traditional tools. The goal is

to implement machine learning to predict the output of Doaks' Decomposition from local parameters so that full domain simulations aren't necessary as well as lowering the computation time required.

2 Methods

2.1 The Data Set

The jet noise data analyzed during for this project was a CFD data set, created by the Cascade Technologies. The jet was simulated using a Large Eddy Simulation (LES), and it is of a mach 0.9, circular, axisymmetric jet. This data set was chosen since the data set is well validated and representative of actual jets. Great care was taken by the team that generated the data and the turbulent boundary layer inside the nozzle was modeled, which is found in real jets but not typically simulated due to complexity. The data from the simulation is statistically stationary, so the mean quantities of the flow don't change with respect to time. 3804 time steps were used, with a non dimensional time of 0.8 separating the snapshots. Visualization tools were used to ensure the validity of the data and the visualizations are included below. The mean velocity profile of the jet is plotted below.

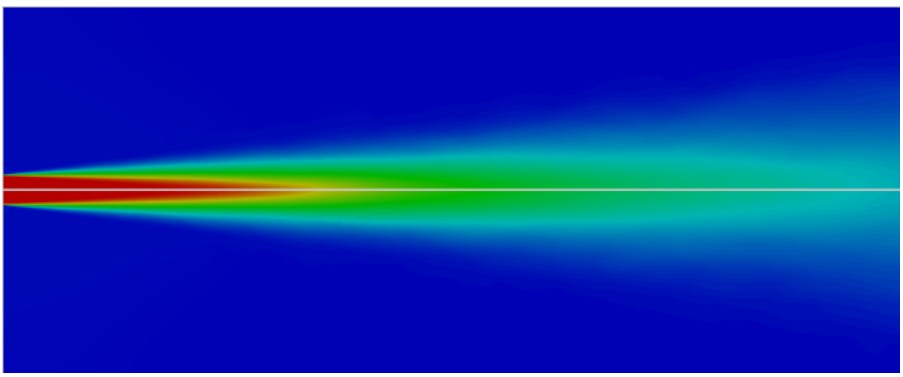

Figure 3: Jet Mean Flow

Figure 3 above verifies that the jet is axisymmetric, as the velocity profile above and below the axis line is the same. The flow velocity is the highest right after it exits the jet, and then as it mixes with the stationary ambient air downstream, the flow

slows. The potential core of the jet is also clearly visible, and it shrinks as the shear layer grows. These are all expected characteristics of jet flow and inspire confidence in the data. Another plot to verify the data is given, this one of the turbulent kinetic energy (TKE), and is given as Figure 4.

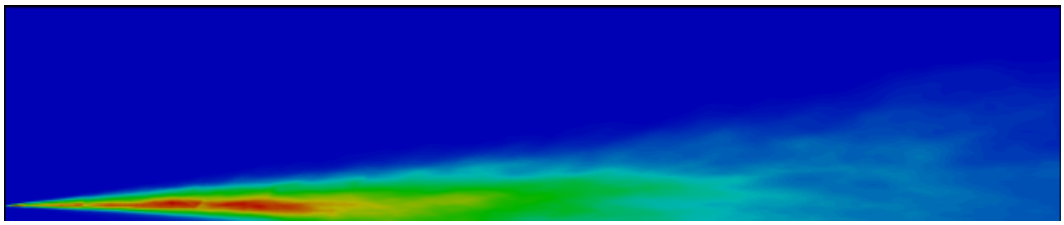

Figure 4: Turbulent Kinetic Energy

The plot above shows the turbulent kinetic energy of the flow, which indicates the strength of the turbulence in a region. The most turbulent region is expected to be the shear layer, which is what is depicted in the plot. The ambient air and the flow coming out of the jet has low TKE and the turbulence decreases in strength after the shear layer as expected. The vorticity in the flow was visualized using an Iso-Surface of the Q-criterion and is shown below. The plot above helps visualize

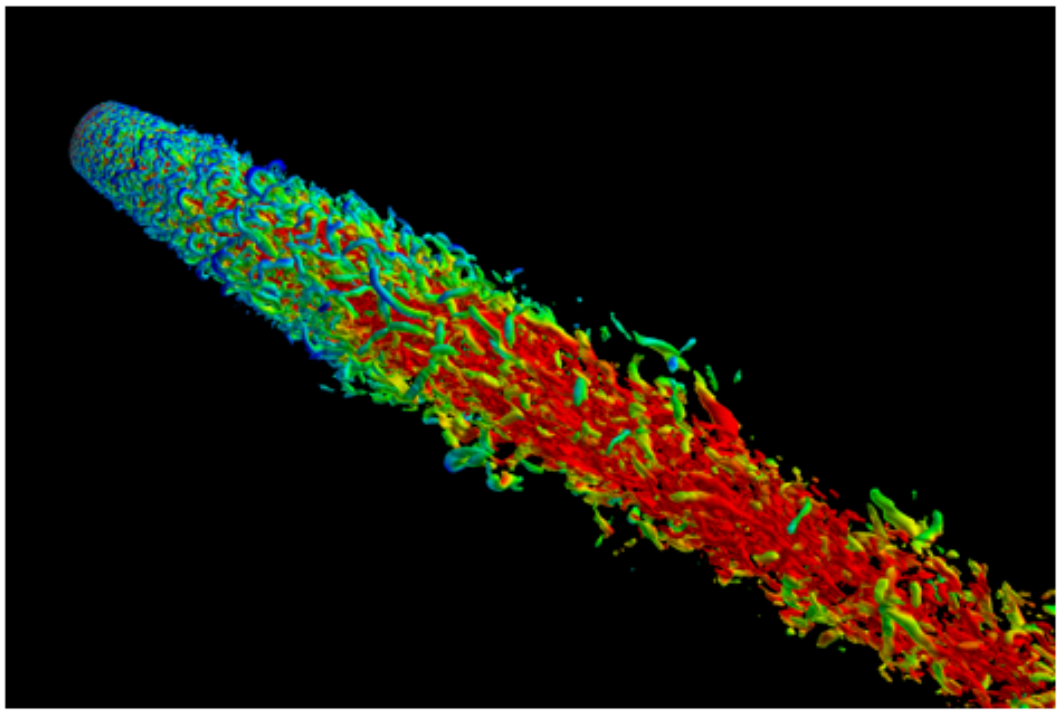

Figure 5: Q-Criterion

the turbulent structures within a flow. The structures close to the jet exit towards the top left corner of the plot are smaller, indicative of the fine-scale turbulence that is expected in that region. Further downstream of the jet, the structures become larger, which are the large scale coherent structures depicted in Figure 1. These plots above help verify that the data is representative of actual jets. For further information on the validity of the simulation, please refer to [1].

The data from the simulation does not have an obvious acoustic component so it must be extracted. This was done by running Doak's Decomposition on the data set. Doak's Decompositon separates the hydrodyamic and acoustic modes, with the acoustic mode given by the Eqaution 1 [14].

$$\nabla^2 \psi'_A = -\frac{1}{c^2}\frac{\partial p'}{\partial t} \tag{1}$$

In the above equation, ψ'_A is the perturbation of the acoustic component of the irrotational momentum density, c is the local speed of sound, and p' is the pressure perturbations [14]. The results of the decomposition, and the data that will be predicted can be visualized in the figure below.

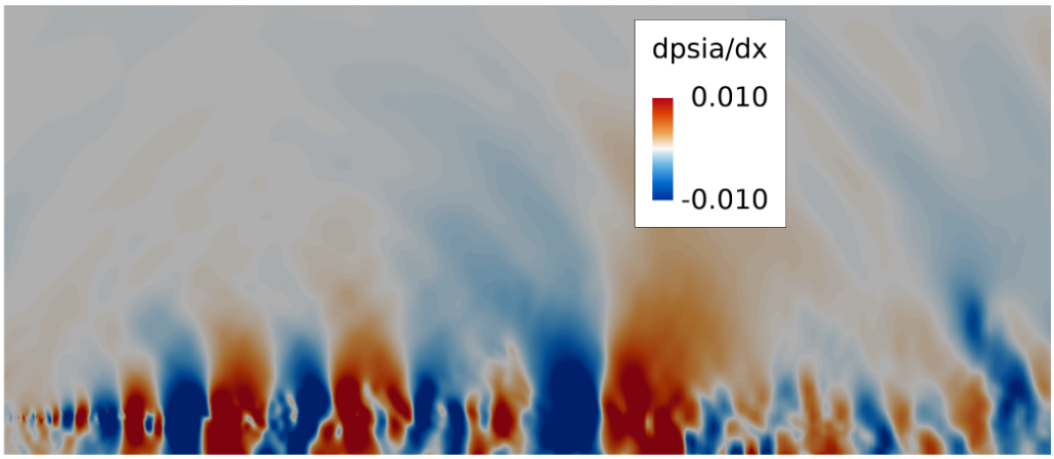

Figure 6: Visualization of Doak's Decomposition

The effects of the velocity (u, v, w), density (ρ) and pressure (p) perturbations on the acoustic response were considered. These were picked since these are vari-

ables that can more easily be determined in experimental setups, and finding these correlations would be the easiest to implement and validate experimentally. The locations that the data was taken from can be seen below in Figure 7.

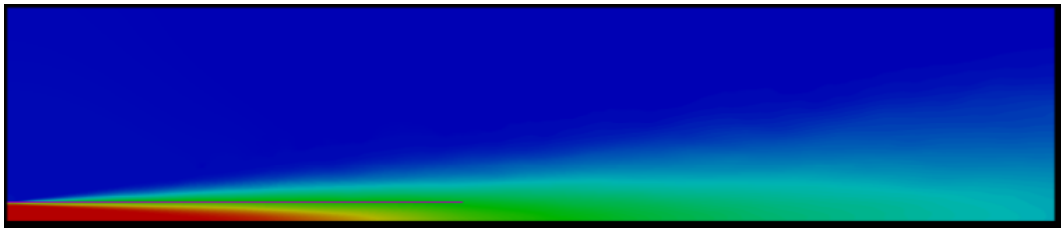

Figure 7: Data Extraction Location

The line in the shear layer indicates that data points that were used to train the neural networks. These data points were chosen to incorporate data in a wide range of flow conditions, as well has to capture the effects of both fine and large scale turbulence.

2.2 Machine Learning Implementation

Machine learning was implemented by using Keras, a library in Python which makes building networks quicker. Fine tuning the network to improve the performance is also easier, as many optimizers and parameters are built into Keras. The first network tested was a simple linear regression model. This was used to see if a linear model could provide reasonable accuracy in capturing the relationships between the perturbations mentioned above and the acoustic component of the flow. The network has five inputs, u', v', w', ρ' and p' and no hidden layers. The ' represents the perturbations from the mean. The neural network is depicted graphically as Figure 8 below.

In the Figure 8, the blue circle is a node, x is an input, w is a weight corresponding to a particular input and $Y(\vec{x})$ is the output as a function of the inputs. $Y(\vec{x})$ in this case is a scalar output. In this case, the input vector consists of u', v', w', ρ' and p'. The output is the acoustic component of the flow The mathematical representation

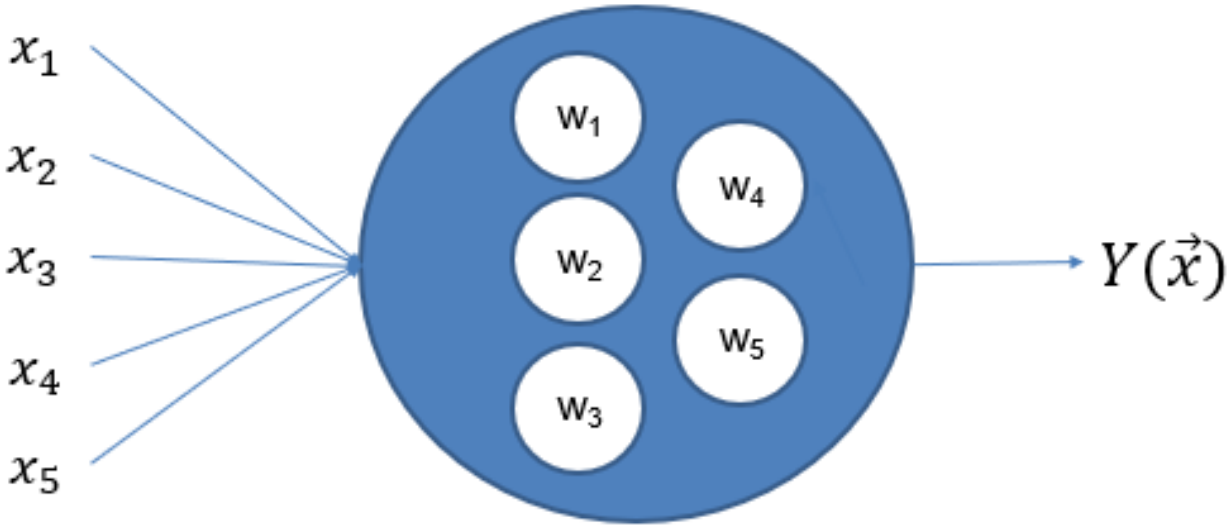

Figure 8: Linear Regression Neural Network

of the above network is given below as equation 2.

$$\begin{pmatrix} x_1 & x_2 & x_3 & x_4 & x_5 \end{pmatrix} \begin{pmatrix} w_1 \\ w_2 \\ w_3 \\ w_4 \\ w_5 \end{pmatrix} = x_1w_1 + x_2w_2 + x_3w_3 + x_4w_4 + x_5w_5 = Y(\vec{x}) \quad (2)$$

From equation 2 above, it is clear to see that this neural network takes the form of a linear regression model. For this project, the linear regression model is given by equation 3 below.

$$\begin{pmatrix} u' & v' & w' & \rho' & p' \end{pmatrix} \begin{pmatrix} w_1 \\ w_2 \\ w_3 \\ w_4 \\ w_5 \end{pmatrix} = u'w_1 + v'w_2 + w'w_3 + \rho'w_4 + p'w_5 = Y(\vec{x}) \quad (3)$$

The model takes the inputs at a specific grid points in the simulation, and the acoustic response at that grid point is the target that the networks predicts. The

grid points chosen were along the shear layer and extended about 5 jet diameters down stream in order to get a wide range of data.The inputs were scaled locally from -1 to 1 since this range limits the size of the weights, which makes tuning the network easier. This also allows the scale of a change in input to be represented. For example, a change from 9 to 10 of some input is a larger relative change than a change from 99 to 100. By scaling the data, this change of 1 would have a larger representation for the first case, and the magnitude of the parameters is not relevant, but only their variance. This model was chosen as a baseline to determine how effective a linear model was in predicting the acoustic response at a given node. The region of interest was the shear layer as this is where the flow is turbulent, and all data to train and test the model was taken from this turbulent region.

The model trains on 75,000 training examples and updates the weights in the network every 1000 training examples. The error used for this training was mean absolute error, where the error is given taken the absolute value of the difference in the predicted and actual value. The weights are updated based on the error in the predictions. The weights are randomly initialized and updated by taking the derivative of the error with respect to the weights as shown in equation 4.

$$\nabla w_{ij} = -\eta \frac{\partial E}{\partial w_{ij}} \tag{4}$$

Where w_{ij} is the weight being updated, E is the error in the prediction, and η is the learning rate, a parameter that is tweaked based on the problem. Each weight is updated at the same time. The error (loss) is calculated using equation 5

$$MAE = \frac{\sum_{i=1}^{n} |y_i - y_p|}{n} \tag{5}$$

Where MAE is the mean absolute error, i is the training example, n is the number of training examples, y_i is the actual value of the training example, and y_p is the value predicted by the network.

This process is repeated for 25 epochs, where an epoch is the whole batch of training data. The model trains on the whole training set multiple times, once per each epoch. After each epoch, the network is tested on a validation set of data, data that the network hasn't trained on, to ensure that the network is not over fitting to the training set. These results give a baseline to compare more complicated models to. The second neural network used was a multilayer perceptron (MLPs). The difference between MLPs from the linear regression model is that there are more nodes per layer as well as more layers between the inputs and output. A depiction of an MLP network is shown below as Figure 9.

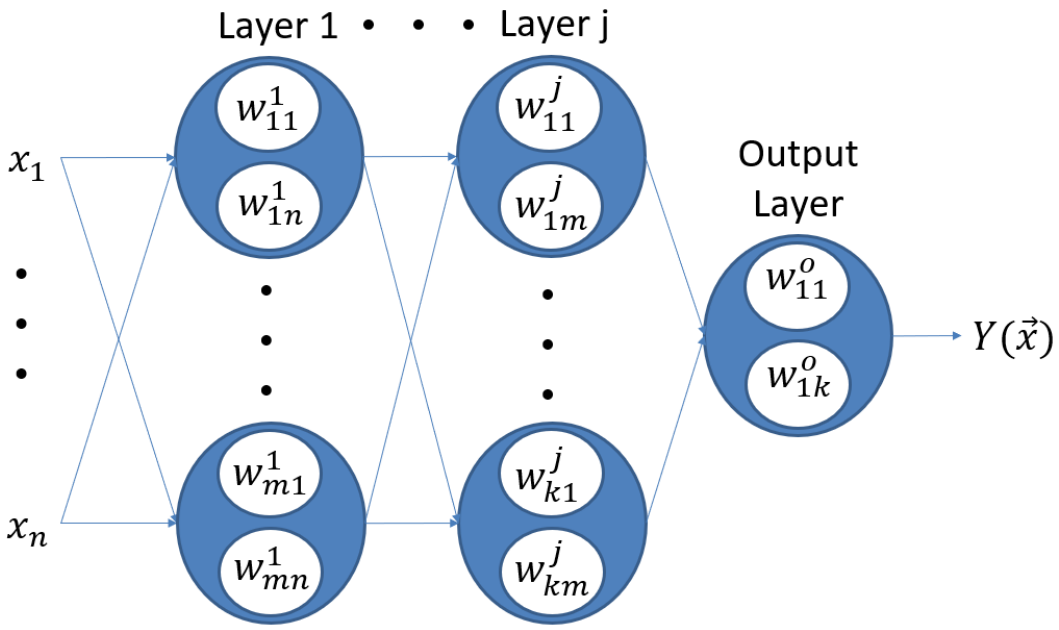

Figure 9: Multi-Layer Perceptron Network

In Figure 9, n is the number of inputs, x represents a specific input, w represents a weight corresponding either an input or a node output, the first subscript is the node in a specific layer, the second subscript is what the weight affects in the previous layer, either an input or node, and the superscript is the layer the node belongs to. Each blue circle is a node and $Y(\vec{x})$ is the output as a function of the inputs.

The benefit of this network is that it can capture non-linearities in the relationship between the input and output, due to the multiplication of values in multiple layers. Another difference is that the output of each node is put through an activation function, which is a function that modifies the input of a node within the desired range. For this problem, the tanh activation function was chosen, so the output of each node is passed through a hyperbolic tangent function. Tanh was chosen since $tanh(x)$ has a range of -1 to 1. This network is expected to better fit the data because of the non linear nature of turbulence. The same 5 inputs that were used before were used for this network as well as the same scaling. The same number of training examples, validation examples, and epochs were used as before. Different MLP networks were tested, with various configurations and the results from all of the models are shown in the next section.

3 Results

3.1 Linear Regression

A linear regression model was created to analyze the linear correlations between u', v', w', ρ', p' and the acoustic component of the flow. The model was trained on 75,000 training examples for 25 epochs, or 25 iterations through the training examples. The model was validated on 25,000 examples. The training loss and validation loss are shown below in Figure 10. The figure shows that the network did improve as it was trained for more epochs, going from a loss of about 0.2796 to a loss of 0.1846, with the loss calculate as shown in equation 5. The loss asymptotes however, indicating that the network has reached the best that it can do. The loss itself doesn't reveal much about how well the network did so the accuracy is given below as Figure 11.

From Figure 11, it is clear that the accuracy, even though it improved as the network trained, was never high, not even reaching 1%. This indicates that the model was only able accurately predict a handful of both the training examples and validation

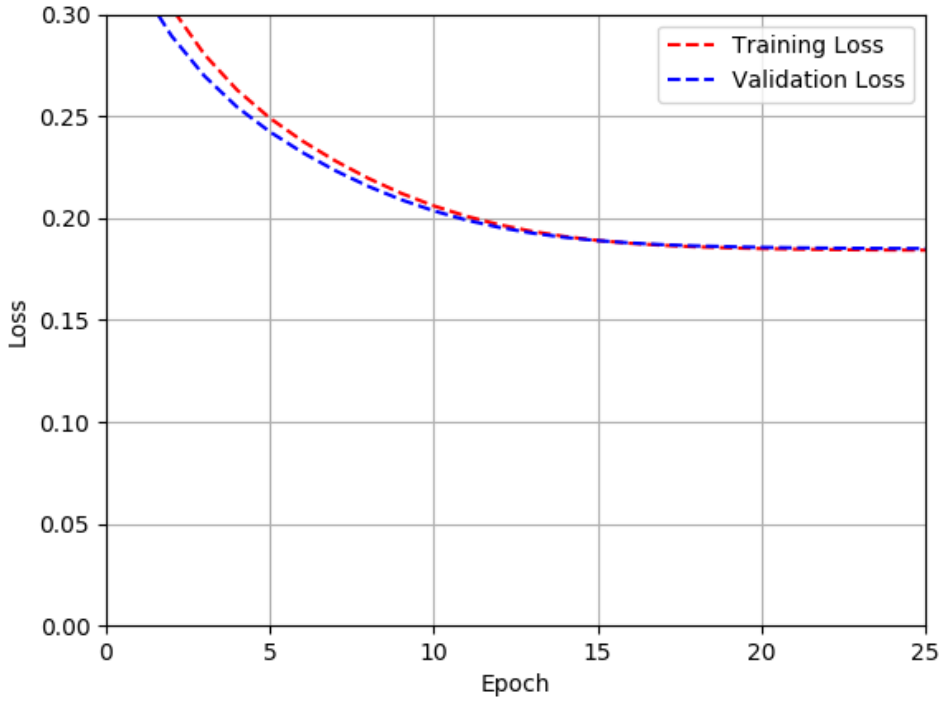

Figure 10: Linear Regression Network Loss

cases, 0.016% and 0.011% respectively. This low of an accuracy indicates that the model is not network isn't learning the system and its responses and this is somewhat expected, since a linear model would have difficulties accurately predicting non-linear responses.

3.2 Multi-Layer Perceptron Network

A MLP network was used next in an attempt to learn the non-linearities of the problem. The same process was used as before, but as the network is larger, it takes longer to train. The loss during training is plotted in Figure 12. The figure shows a similar trend with the training loss decreasing for the first few epochs and then leveling off, at a value of 0.1844, and the validation loss ends at 0.1843, both of which are slightly lower than the previous network. For this network the convergence on the loss was quicker that in the previous case, which indicates that it

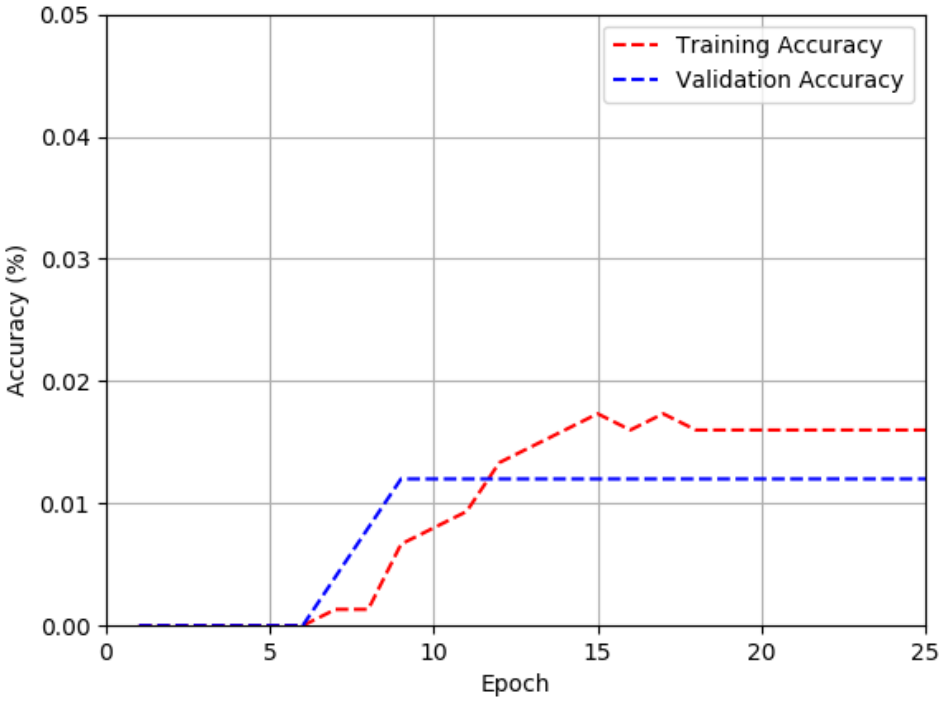

Figure 11: Linear Regression Network Accuracy

captures some of the relationships quicker than the other network. The loss however is not significantly different. The accuracy is also plotted and given in Figure 13. The accuracy also rises as the data has been trained on for more epochs, however it never reaches significant values, with the training accuracy asymptoting at 0.015% and the validation accuracy never growing beyond 0.008%. These are again not significant accuracies and can be attributed to chance. This is a surprising result, as the MLP would be expected to be able to learn the non-linearities better than the regression model, but it seems to achieve the same if not slightly worse results.

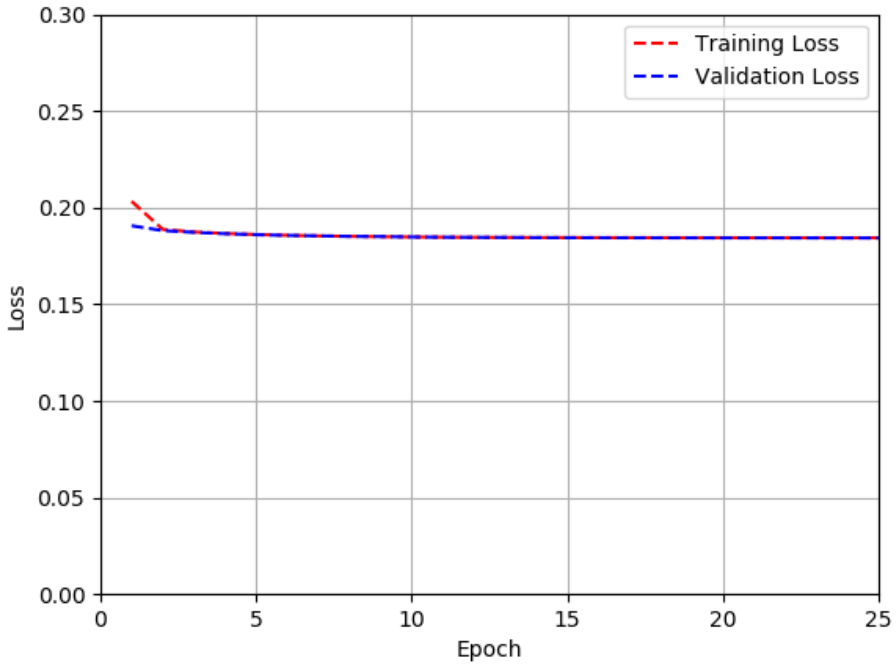

Figure 12: Multi-Layer Perceptron Network Loss

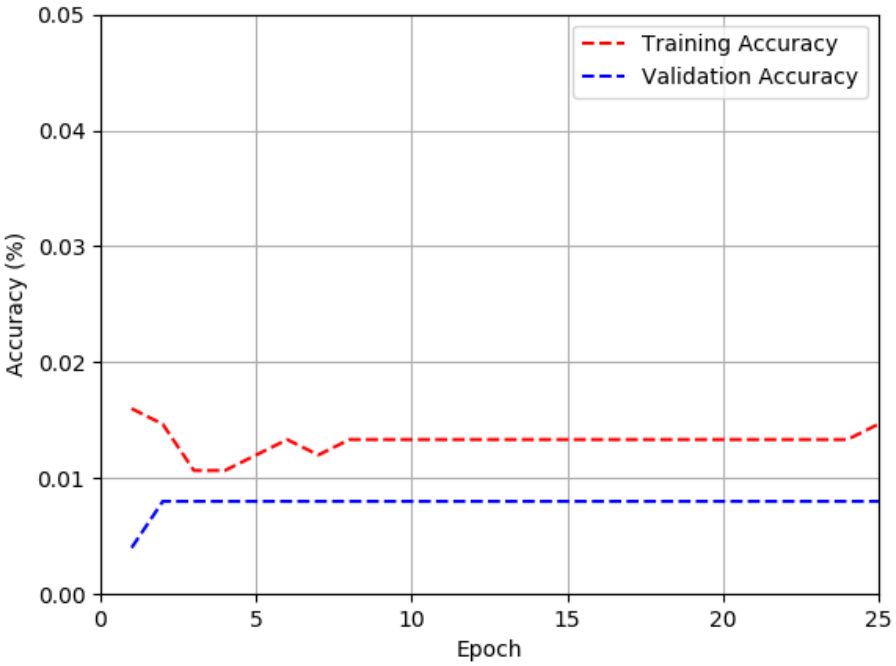

Figure 13: Multi-Layer Perceptron Network Accuracy

To see if more training examples were needed, the MLP network was retrained on 750,000 training examples with 250,000 validation examples. The weight were up-

dated every 5000 examples to keep the computation time low. Figures 14 and 15 show the loss and accuracy respectively. The loss leveled off to similar values as before, 0.1843 for the training set and 0.1844 for the validation set. The accuracy actually decreased, leveling off at 0.00773% for the training examples and 0.0060% on the validation cases. This mean that even with 10 times the number of training examples, the model was not able to improve in performance, and actually lowered in accuracy.

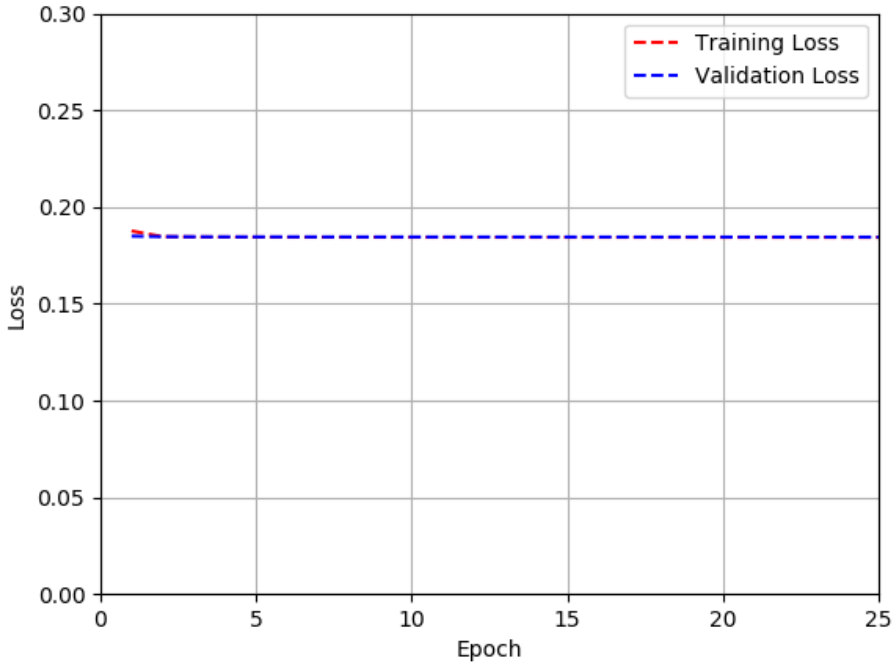

Figure 14: MLP Network Loss-More Examples

This points to the input parameters not correlating well to the acoustic response of the jet. To check this, v' and w' were left out and the network was retrained on 75,000 training examples. The loss and accuracy are given in Figures 16 and 17. The figures show that even removing 2 of the input parameters doesn't really affect the performance of the network.

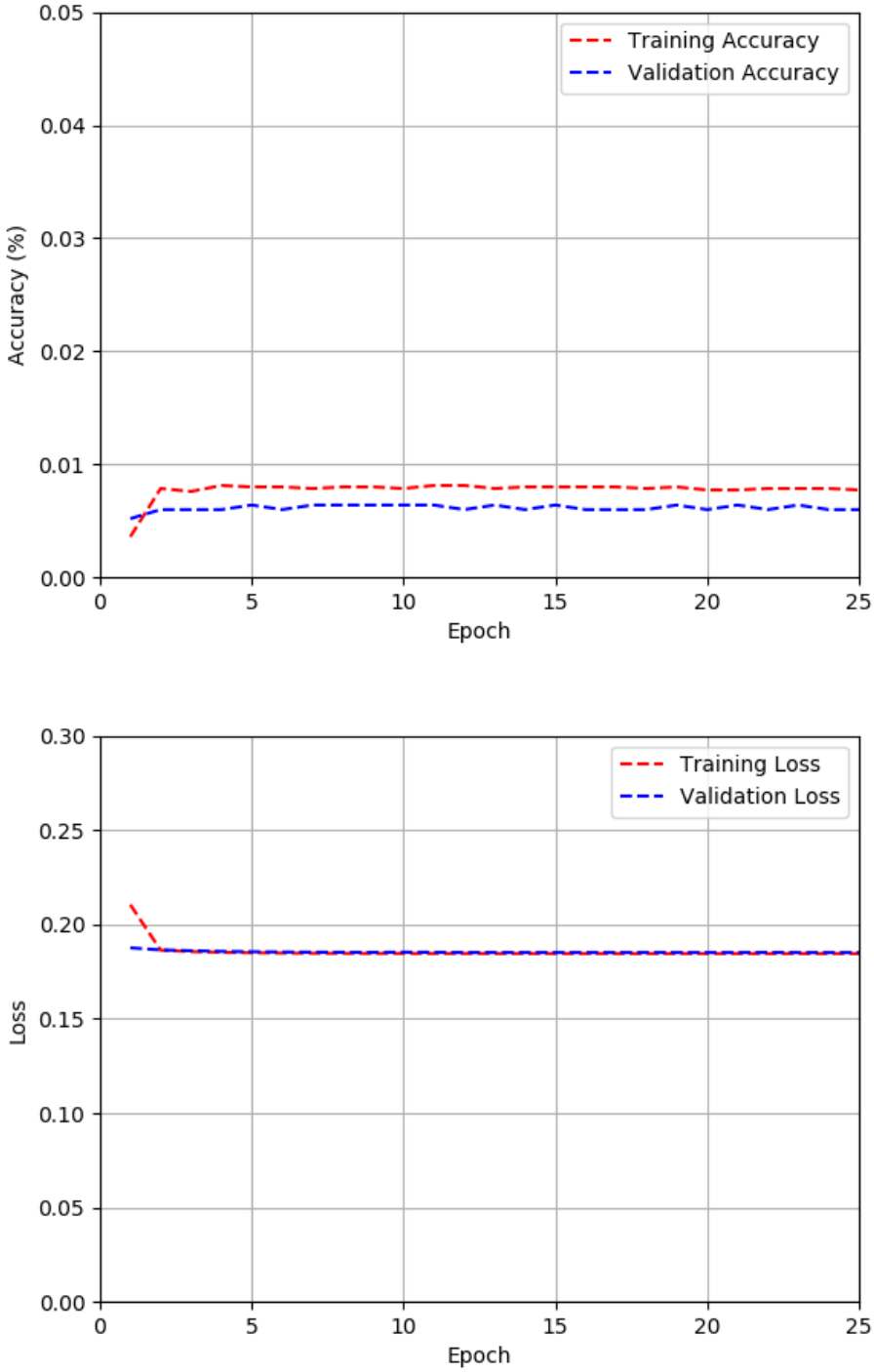

Figure 16: MLP Network Loss: Inputs of u', ρ' and p'

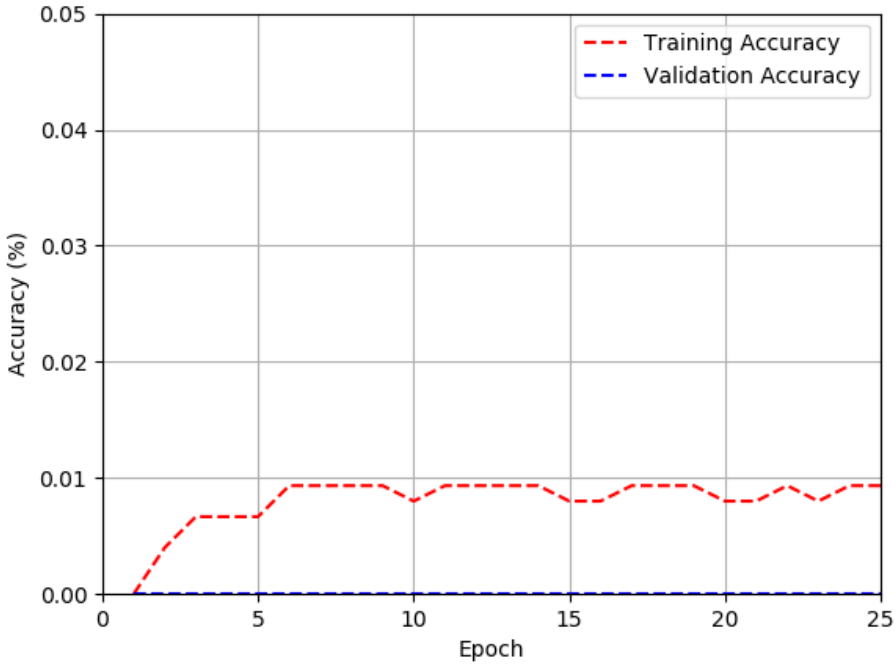

Figure 17: MLP Network Accuracy: Inputs of u', ρ' and p'

3.3 Analysis of Results

The low accuracy of the predictions indicates a few things. The first response to the low accuracy would be the model was not able to capture the non-linearities of the problem. This however is unlikely for a couple of reasons, the first being that the size of the MLP network means that there are many multiplications of weights and inputs, so many orders of non-linearities would be modeled. The second is that the the MLP network did not do much better than the linear regression model, meaning that any non-linearities that were accounted for did not help with the prediction. This points to the input parameters selected not being good indicators of the acoustic response within the flow field. This is supported by the fact the removing the velocity perturbation did not hurt the performance of the model, especially the loss, greatly. This low accuracy of the networks can be explained if the inputs don't correlate with the acoustic component, as the neural network can't learn correlations that don't exist. These issues can be explored in later works with more training, different inputs, and networks.

3.4 Future Work

The issues with the model accuracy can be remedied by using different input parameters. Equation 1 can be referenced to determine which parameters to use for further training. The equation shows the the acoustic component clearly correlates with $\frac{\partial p'}{\partial t}$ so implementing this into the network would yield better accuracy. This can be done in a few ways. One would be to explicitly calculate the time derivative of the pressure perturbation and feed that into the network as an input parameter. Another way would be to use other types of networks. Recurrent neural networks are a class of networks where the nodes are connected temporally. This would allow the network to keep track of inputs over time, which would allow it to learn the relationship between $\frac{\partial p'}{\partial t}$ and the acoustic response. An examples of such a network is a Long-Short Term Memory (LSTM) network. Equation 1 also shows that the local speed of sound is also an important factor. Again, this could be explicitly

fed as an input to the network or the ratio of specific heat, gas constant and local temperature could be given, since those 3 parameters determine the speed of sound. These are some ways in which the model can be altered so that it can accurately predict the output of Doak's Decomposition.

4 Conclusion

4.1 Summary

The work done over the course of this project consisted of analyzing simulation jet noise data using machine learning techniques in order to correlate various variables to the acoustic component of the flow. The variables considered were velocity perturbations, u', v' and w', density perturbations, ρ', and pressure perturbations p'. The data was taken from a well validated mach 0.9 jet, where the boundary layer inside the jet was turbulent, which closely models real jets. The acoustic component of the flow was extracted using Doak's decomposition while the input parameters were taken directly from the data set and all the data at each node was scaled between -1 to 1. Both a linear regression model and multi-layer perceptron model were trained on the data set but both were unable to find strong correlations between the input variables and the acoustic component of the flow, as evidenced by the low accuracy over all the training and validation cases. Even when the velocity perturbations were removed as inputs, the MLP network loss did not change significantly indicating that those inputs did not strongly correlate to the output. Some ways to remedy this include using $\frac{\partial p'}{\partial t}$ and c as inputs, as well as changing the type of network to one that can keep track of inputs over time.

References

[1] Guillaume A Bres, Vincent Jaunet, Maxime Le Rallic, Peter Jordan, Tim Colonius, and Sanjiva K Lele. Large eddy simulation for jet noise: the importance of getting the boundary layer right. In *21st AIAA/CEAS aeroacoustics conference*, page 2535, 2015.

[2] Marie Camilleri, Joël Ducourneau, and Rémy Pujol. The number one enemy: Noise !, Jun 2017.

[3] Naval Research Advisory Committee et al. Report on jet engine noise reduction. *URL: http://www. nrac. navy. mil/docs/2009_FINAL_Jet_Noise_Report_4-26-09. pdf*, 2009.

[4] S Cj Crow and FH Champagne. Orderly structure in jet turbulence. *Journal of Fluid Mechanics*, 48(3):547–591, 1971.

[5] Felix Faber, Alexander Lindmaa, O Anatole von Lilienfeld, and Rickard Armiento. Crystal structure representations for machine learning models of formation energies. *International Journal of Quantum Chemistry*, 115(16):1094–1101, 2015.

[6] Amir Barati Farimani, Joseph Gomes, and Vijay S Pande. Deep learning the physics of transport phenomena. *arXiv preprint arXiv:1709.02432*, 2017.

[7] Elham Heidari, Mohammad Amin Sobati, and Salman Movahedirad. Accurate prediction of nanofluid viscosity using a multilayer perceptron artificial neural network (mlp-ann). *Chemometrics and intelligent laboratory systems*, 155:73–85, 2016.

[8] Michael James Lighthill. On sound generated aerodynamically i. general theory. *Proceedings of the Royal Society of London. Series A. Mathematical and Physical Sciences*, 211(1107):564–587, 1952.

[9] Bernard Marr. A short history of machine learning – every manager should read, Mar 2016.

[10] Jean Rabault, Ulysse Reglade, Nicolas Cerardi, Miroslav Kuchta, and Atle Jensen. Deep reinforcement learning achieves flow control of the 2d karman vortex street. *arXiv preprint arXiv:1808.10754*, 2018.

[11] Unnikrishnan Sasidharan Nair. *Jet noise source localization and identification*. PhD thesis, The Ohio State University, 2017.

[12] Christopher KW Tam. Supersonic jet noise. *Annual review of fluid mechanics*, 27(1):17–43, 1995.

[13] Christopher KW Tam, K Viswanathan, KK Ahuja, and J Panda. The sources of jet noise: experimental evidence. *Journal of Fluid Mechanics*, 615:253–292, 2008.

[14] S Unnikrishnan and Datta V Gaitonde. Acoustic, hydrodynamic and thermal modes in a supersonic cold jet. *Journal of Fluid Mechanics*, 800:387–432, 2016.

Appendices

Appendix A: Machine Learning Implementation

A Linear Regression Code

```
import pandas as pd
import numpy as np
from keras.models import Sequential
from keras.layers import Dense
from keras import losses
import keras
import math
import matplotlib.pyplot as plt
from matplotlib import interactive
from keras.utils import plot_model
# confirm TensorFlow sees the GPU
from keras.callbacks import History
hist = History()
from tensorflow.python.client import device_lib
print(device_lib.list_local_devices())


print('Getting data')
filename = "scaled_data.txt"
datatemp = np.loadtxt(filename)
print(datatemp)
print(datatemp[0,0:5])
print(datatemp[0,5])
#print(datatemp)
#datatest = np.arange(9).reshape((3,3))
```

```
#print(datatest)
#np.random.shuffle(datatest)
#print(datatest)
np.random.shuffle(datatemp)
print('Building Network')
model=Sequential()
model.add(Dense(1,input_dim=5,activation='linear'))
model.compile(loss=losses.mean_absolute_error,...
...optimizer=keras.optimizers.Adam(decay=0.0),metrics=['accuracy'])
datapoints=100000
xtrain=datatemp[0:datapoints,0:5]
ytrain=datatemp[0:datapoints,5]
print('Training Model')
hist=model.fit(xtrain,ytrain,validation_split=0.25,batch_size=1000,epochs=25,...
...shuffle=True,verbose=2)
tloss = hist.history['loss']
tacc = hist.history['acc']
tacc = [x * 100 for x in tacc]
vacc = hist.history['val_acc']
vacc = [x * 100 for x in vacc]
vloss = hist.history['val_loss']
epochs = range(1, len(tloss) + 1)

plt.figure(1)
plt.plot(epochs,tloss,'r--')
plt.plot(epochs,vloss,'b--')
plt.grid(b=True,which='both')
plt.xlabel('Epoch')
plt.ylabel('Loss')
plt.axis([0.0,25.0,0.0,0.3])
```

```
plt.legend(['Training Loss','Validation Loss'])
interactive(True)
plt.show()
#plt.title('Linear Regression Loss')

plt.figure(2)
plt.plot(epochs,tacc,'r--')
plt.plot(epochs,vacc,'b--')
plt.grid(b=True,which='both')
plt.xlabel('Epoch')
plt.ylabel('Accuracy (%)')
plt.axis([0.0,25.0,0.0,5e-2])
plt.legend(['Training Accuracy','Validation Accuracy'])
plt.show()
interactive(False)
#plot_model(model, to_file='Linear Network.png')
```

B MLP Code

```
import pandas as pd
import numpy as np
from keras.models import Sequential
from keras.layers import Dense
from keras import losses
import keras
import math
from keras.callbacks import History
hist = History()
from tensorflow.python.client import device_lib
import matplotlib.pyplot as plt
```

```
from keras.utils import plot_model
from matplotlib import interactive
vars=5
print(device_lib.list_local_devices())
print('Getting data')
filename = "scaled_data.txt"
datatemp = np.loadtxt(filename)
print(datatemp)
print(datatemp[0,0:5])
print(datatemp[0,5])
np.random.shuffle(datatemp)
print('Building Network')
model=Sequential()
model.add(Dense(64,input_dim=vars,activation='tanh'))
model.add(Dense(64,activation='tanh'))
model.add(Dense(64,activation='tanh'))
#model.add(Dense(64,activation='tanh'))
#model.add(Dense(64,activation='tanh'))
model.add(Dense(1,activation='tanh'))
model.compile(loss=losses.mean_absolute_error,...
...optimizer=keras.optimizers.SGD(decay=0.0),metrics=['accuracy'])
model.summary()
datapoints=100000
xtrain=datatemp[0:datapoints,0:5]
ytrain=datatemp[0:datapoints,5]
xpred=datatemp[datapoints:(datapoints+10),0:5]
ypred=datatemp[datapoints:(datapoints+10),5]
print('Training Model')
hist=model.fit(xtrain,ytrain,validation_split=0.25,batch_size=5000,epochs=25,...
...shuffle=True,verbose=2)
```

```
tloss = hist.history['loss']
tacc = hist.history['acc']
tacc = [x * 100 for x in tacc]
vacc = hist.history['val_acc']
vacc = [x * 100 for x in vacc]
vloss = hist.history['val_loss']
epochs = range(1, len(tloss) + 1)
plt.figure(1)
plt.plot(epochs,tloss,'r--')
plt.plot(epochs,vloss,'b--')
plt.grid(b=True,which='both')
plt.xlabel('Epoch')
plt.ylabel('Loss')
plt.axis([0.0,25.0,0.0,0.3])
plt.legend(['Training Loss','Validation Loss'])
interactive(True)
plt.show()
#plt.title('Linear Regression Loss')

plt.figure(2)
plt.plot(epochs,tacc,'r--')
plt.plot(epochs,vacc,'b--')
plt.grid(b=True,which='both')
plt.xlabel('Epoch')
plt.ylabel('Accuracy (%)')
plt.axis([0.0,25.0,0.0,5e-2])
plt.legend(['Training Accuracy','Validation Accuracy'])
plt.show()
interactive(False)
#plot_model(model, to_file='Linear Network.png')
```